

## AMENDMENTS TO THE CLAIMS

1. (Previously presented) In a computing device having at least one plug-in module that extends the functionality of a Web browser, a method of identifying a plug-in module that generated a failure, comprising:

in response to receiving notice of a failure, obtaining selected contents of memory of said computing device created at the time of the failure;

generating a failure signature that is characteristic of the plug-in module that generated the failure;

comparing said failure signature with one or more failure signatures generated by known plug-in modules; and

allowing a user to disable the plug-in module that is the source of the failure instead of a category of plug-in modules.

2. (Original) The method as recited in Claim 1, further comprising:

if the failure signature is characteristic of a known plug-in module, determining if the known plug-in module has an update that does not generate a failure.

3. (Previously presented) The method as recited in Claim 2, further comprising:

if the known plug-in module has an update that does not generate failures, informing a user of the availability of the update.

4. (Canceled)

5. (Original) The method as recited in Claim 1 wherein obtaining the contents in memory of said computing device at the time of the failure includes obtaining a minidump file.

6. (Previously presented) The method as recited in Claim 1 wherein generating a failure signature from the contents of memory that is characteristic of the plug-in module that generated the failure includes:

identifying a library that was executing at the time of the failure;  
determining the plug-in module that uses said library; and  
identifying the application that interacts with the plug-in module that uses said library.

7. (Original) The method as recited in Claim 6 wherein identifying the library that was executing at the time of the failure includes searching a minidump file.

8. (Previously presented) The method as recited in Claim 6 wherein determining the plug-in module that uses the library includes searching a system registry for associations between plug-in modules and libraries.

9. (Previously presented) The method as recited in Claim 6 wherein identifying the application that interacts with the plug-in module includes searching a system registry for associations between applications and plug-in modules.

10. (Previously presented) A computer-readable medium bearing computer-executable instructions that, when executed, carry out a method of identifying a plug-in module that generated a failure, comprising:

in response to receiving notice of a failure, obtaining selected contents of memory of said computing device created at the time of the failure;

generating a failure signature that is characteristic of the plug-in module that generated the failure;

comparing said failure signature with one or more failure signatures generated by known plug-in modules; and

allowing a user to disable the plug-in module that is the source of the failure instead of a category of plug-in modules.

11. (Original) The computer-readable medium as recited in Claim 10, further comprising:

if the failure signature is characteristic of a browser plug-in module, determining if the known plug-in module has an update that does not generate a failure.

12. (Previously presented) The computer-readable medium as recited in Claim 11, further comprising:

if the known plug-in module has an update that does not generate failures, informing a user of the availability of the update.

13. (Canceled)

14. (Original) The computer-readable medium as recited in Claim 10 wherein obtaining the contents in memory of said computing device at the time of the failure includes obtaining a minidump file.

15. (Previously presented) The computer-readable medium as recited in Claim 10 wherein generating a failure signature from the contents of memory that is characteristic of the plug-in module that generated the failure includes:

identifying the segment of code that was executing at the time of the failure; and

determining a library that contains said segment of code that was executing at the time of the failure; and

identifying the application that interacts with the plug-in module that uses said library.

16. (Previously presented) The computer-readable medium as recited in Claim 15, further comprising determining if said segment of code that was executing at the time of the failure includes searching a minidump file.

17. (Previously presented) The computer-readable medium as recited in Claim 16 wherein determining if said segment of code is from a library used by a plug-in module to a Web browser includes searching a system registry for associations between plug-in modules and libraries.

18. (Previously presented) The computer-readable medium as recited in Claim 15 wherein identifying the segment of code that was executing at the time of the failure includes searching a system registry for associations between applications and plug-in modules.

19. (Currently amended) A computer-readable medium bearing computer-executable instructions which, when executed:

identifies plug-in modules used in conjunction with a Web browser;

identifies automatically and without further user input, a plug-in module that generated a failure based on a failure signature;

displays automatically and without further user input, a graphical user interface that lists the plug-in modules used in conjunction with a Web browser; and

supports disabling the specific plug-in module used in conjunction with the Web browser that generated the failure instead of a category of plug-in modules.

20. (Previously presented) The computer-readable medium recited in Claim 19 wherein identifying one or more plug-in modules used in conjunction with the Web browser

includes searching a system registry for associations between plug-in modules and the Web browser.

21. (Previously presented) The computer-readable medium recited in Claim 19 wherein the graphical user interface indicates whether each plug-in module used in conjunction with the Web browser that generated a failure may be updated.

LAW OFFICES OF  
CHRISTENSEN O'CONNOR JOHNSON KINDNESS<sup>PLLC</sup>  
1420 Fifth Avenue  
Suite 2800  
Seattle, Washington 98101  
206.682.8100